



## Supervision des réseaux et services pair à pair : application à la plate-forme JXTA

Guillaume Doyen, Emmanuel Nataf, Olivier Festor

### ► To cite this version:

Guillaume Doyen, Emmanuel Nataf, Olivier Festor. Supervision des réseaux et services pair à pair : application à la plate-forme JXTA. CFIP 2006 - Colloque Francophone sur l'Ingénierie des Protocoles, Eric Fleury and Farouk Kamoun, Oct 2006, Tozeur, Tunisie. pp.12. inria-00113390

**HAL Id: inria-00113390**

**<https://inria.hal.science/inria-00113390>**

Submitted on 20 Nov 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Supervision des réseaux et services pair à pair : application à la plate-forme JXTA

**Guillaume Doyen — Emmanuel Nataf — Olivier Festor**

LORIA – INRIA Lorraine  
Campus Scientifique  
BP 239  
54506 Vandoeuvre-lès-Nancy Cedex  
FRANCE  
Guillaume.Doyen@loria.fr

---

*RÉSUMÉ. Le déploiement d'applications pair à pair (P2P) dans des environnements où la qualité de service doit être garantie passe par l'intégration de mécanismes de supervision. Nous présentons ici une infrastructure de supervision pour la plate-forme JXTA. Celle-ci repose sur un modèle de l'information de gestion générique pour la gestion des réseaux et services P2P que nous avons conçu préalablement. Nous présentons la manière dont nous avons instancié ce modèle à travers une architecture de supervision reposant sur un modèle gestionnaire/agent qui nous permet d'obtenir une vue globale des éléments d'une communauté P2P JXTA ainsi que de son évolution.*

*ABSTRACT. The deployment of P2P applications in environments where the quality of offered services must be ensured requires the integration of management mechanisms. In this paper, we present a management infrastructure for the JXTA P2P framework. It relies on a generic management information model for P2P networks and services we designed previously. We show the way we instantiated this model through a manager-agent model which makes it possible to get a global view of elements of a JXTA P2P community as well as its evolution.*

*MOTS-CLÉS : Pair à pair, supervision de réseaux, CIM, Jxta, JMX*

*KEYWORDS: Peer-to-peer, network management, CIM, Jxta, JMX*

---

## 1. Introduction

Le modèle P2P est un modèle distribué où les entités appelées pairs jouent le double rôle de client et serveur et interagissent afin d'offrir à une communauté un service de manière décentralisée. Cette décentralisation des ressources et des services présente de nombreux avantages qui repoussent les limites induites par le modèle client-serveur. Parmi celles-ci, on compte (1) une puissance de calcul qui dépasse celle des infrastructures centralisées existantes, (2) la tolérance aux fautes, (3) l'équilibre du trafic et de la charge attribuée à chaque participant et (4) la réduction des coûts liés à la mise en œuvre des équipements.

Toutes ces bonnes propriétés expliquent l'attention particulière qui est portée au modèle P2P. Si les premières applications de ce modèle étaient exclusivement liées à l'échange souvent illégal de données soumises à des droits d'auteur, c'est maintenant l'ensemble des services réseaux qui peuvent être repensés pour être déployés selon une approche P2P. Ce modèle ouvre donc de nouveaux horizons aux applications réseaux déployées, non seulement chez des particuliers, mais aussi dans le cadre plus formel des entreprises, administrations et universités. Ainsi, cette nouvelle utilisation du modèle P2P, dans des environnements contraints où la qualité du service offert est primordiale, nécessite la mise en place d'une infrastructure de supervision qui puisse surveiller et contrôler les services reposant sur ce modèle.

Les approches de gestion actuelles sont conçues pour des services à caractère centralisé et s'adaptent mal aux propriétés du modèle P2P. De nouvelles approches de supervision doivent ainsi être conçues et éprouvées. Le travail présenté ici s'inscrit dans cette démarche et concerne : (1) la proposition d'un modèle de l'information de gestion pour les réseaux et services pair à pair, (2) sa validation à travers une instantiation sur différentes infrastructures P2P existantes qui reposent sur des modèles différents, et (3) sa mise en œuvre sur la plate-forme P2P JXTA, ce dernier point s'inscrivant dans le cadre du projet RNRT Safari<sup>1</sup>.

Le plan de cet article est le suivant. Avant de présenter nos travaux, nous nous intéressons dans la section 2 à l'ensemble des contributions actuelles qui sont relatives à la supervision des services P2P. Ensuite, section 3, nous présentons le modèle de l'information de gestion générique que nous avons conçu ainsi que les différentes instantiations que nous avons faites pour valider sa généricité. La section 4 s'intéresse plus particulièrement au travail conduit sur la plate-forme JXTA, de la spécification d'un modèle de l'information dédié, à la mise en œuvre d'une infrastructure de supervision complète. Enfin, la section 5 présente les conclusions de ce travail ainsi que les travaux à venir.

---

1. <https://safari-rnrt.rd.francetelecom.com>

## 2. Etat de l'art

On compte actuellement plusieurs catégories de travaux qui portent sur la supervision des réseaux P2P. Celles-ci se différencient par la nature du problème, induit par le modèle P2P, qu'elles traitent.

L'utilisation de **modèles économiques** repose sur des modèles de marché issus des domaines de l'économie et des finances. Ils sont particulièrement adaptés aux applications de partage de contenu et exploitent la théorie des jeux [GOL 01], les notions de monopole et concurrence [ANT 02] ou le dilemme du prisonnier [FEL 04]. Leur utilisation permet entre autres la différenciation du service offert aux pairs suivant leur participation, évitant ainsi les pairs "profiteurs" [BUR 03]. Conjointement aux modèles économiques, on trouve des systèmes incitatifs fondés sur **la confiance et la réputation**. Ceux-ci définissent et instancient : (1) une métrique d'évaluation de la confiance [GUP 03, XIO 02], (2) une infrastructure de stockage et d'accès aux données qui stocke l'ensemble des données relatives à l'évaluation des pairs qui entrent dans le cadre de la métrique choisie [ABE 01], et (3) un mécanisme de sécurité qui puisse faire face aux pairs mal intentionnés [SCH 03]. Enfin, la majorité des applications P2P définissent une topologie fondée sur un **réseau virtuel** construit au-dessus des réseaux physiques et logiques. Le maintien de cette topologie dans un bon état de fonctionnement est crucial pour les services qu'ils supportent. Plusieurs travaux proposent des solutions pour les superviser : Cyclon [VOU 05] est une infrastructure de gestion de la topologie des réseaux P2P non structurés qui repose sur un algorithme de *shuffle*. Ensuite, les travaux de Cooper et al. [COO 05] proposent, au sein d'une communauté, de casser régulièrement les connexions entre les pairs en fonction de leur charge. Enfin, AVP (*Active Virtual Peer*), proposée par Kouroulis et al. [KOU 03], ne propose non pas une manière de construire une topologie non structurée efficace, mais plutôt la manière d'utiliser au mieux une topologie donnée.

Conjointement à ces travaux on trouve l'approche duale qui est l'utilisation du P2P pour la gestion de réseaux. En particulier, Konstantinou [KON 03] s'est penché sur la réalisation d'une infrastructure d'autogestion reposant sur une infrastructure P2P pour la supervision des réseaux et services.

Les approches présentées dans cette section reposent essentiellement sur l'intégration de mécanismes de supervision au sein du service supervisé. Notre démarche diffère de ces dernières principalement sur deux points qui sont, d'une part, la séparation effective entre les plans de service et plan de supervision, et d'autre part, l'utilisation et l'adaptation de modèles et mécanismes de gestion existants.

## 3. Proposition d'un modèle de l'information de gestion

La première étape de la conception d'une infrastructure de supervision consiste à définir un modèle de l'information des éléments supervisés. Celui que nous avons conçu pour les réseaux et services P2P [DOY 04] est un schéma d'extension de CIM [BUM 00], un standard du DMTF pour la gestion des réseaux et services.

Un des objectifs que nous nous sommes fixés durant la conception de notre modèle concerne la généricité. Il nous semble important que le modèle de l'information proposé puisse être appliqué à n'importe quelle approche P2P. Par ce biais, une application de gestion peut disposer d'une vue uniforme et cohérente de différentes infrastructures P2P dont elle assure la supervision.

Notre travail a donc consisté à inventorier les concepts communs à toutes les approches P2P, qui sont à considérer dans le cadre de la gestion de réseaux et de services et à les formaliser à travers différents sous-modèles qui concernent l'organisation communautaire, fonctionnelle et topologique, la communication, les ressources, les services et le routage.

### **3.1. *Instanciation du modèle sur différentes infrastructures***

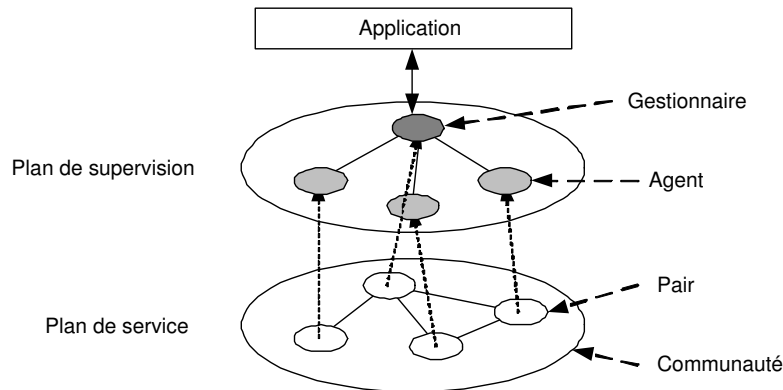
Afin de montrer la généricité de notre modèle, nous l'avons instancié sur plusieurs infrastructures P2P existantes. Pour Chord [STO 01], nous avons montré que notre modèle peut s'instancier sur une table de hachage distribuée (DHT), qui est une infrastructure fondamentale dans les applications P2P actuelles. Sur Pastry [ROW 01] nous montrons que notre modèle est suffisamment générique pour pouvoir prendre en compte les spécificités de n'importe quelle DHT. Enfin, pour JXTA [GON 01], nous avons dû prendre en compte le modèle hybride qu'il utilise, avec des pairs de rendez-vous et des pairs relais.

Ces expériences concluantes ont montré la complétude actuelle de notre modèle face aux services P2P existants. Si de nouveaux services introduisant de nouveaux concepts apparaissaient, l'extensibilité du formalisme CIM permettrait d'y intégrer ces nouveaux concepts. Nous présentons ici l'instanciation que nous avons effectuée sur JXTA en se focalisant sur les aspects inhérents au sous-modèle de l'organisation.

### **3.2. *Proposition d'une infrastructure de déploiement***

Afin de déployer notre modèle de l'information, nous proposons dans un premier temps d'utiliser un modèle standard de type gestionnaire/agent ; l'utilisation d'une infrastructure gestionnaire/agent P2P fait l'objet de travaux complémentaires [DOY 05] qui sortent du sujet de cet article. La figure 1 montre, qu'au sein d'une communauté P2P, un pair est élu gestionnaire de la communauté. Sa fonction est alors d'agréger les différentes informations de gestion issues des agents exécutés par chaque pairs. Chaque agent héberge ainsi une vue locale de notre modèle et le gestionnaire, une vue globale. La vue locale permet de résoudre le problème d'implantation des instances d'objets partagées (comme par ex. les *pipes* qui sont des canaux de communication virtuels).

D'un point de vue du fonctionnement, notre modèle effectue les opérations suivantes : lorsqu'un pair rejoint une communauté, il n'a aucune information sur la présence d'un éventuel gestionnaire et le cas échéant, sur son identité. La première opération qu'il effectue consiste donc à demander aux pairs présents qui est gestionnaire



**Figure 1.** L'organisation du plan de supervision

de ce groupe. Chaque pair du groupe reçoit cette requête et y répond seulement s'il est le gestionnaire. Si un gestionnaire existe, à la réception de ce message, il enregistre le nouveau pair dans la liste de ses agents. Si au bout d'un temps fixé, le nouvel agent ne reçoit aucune réponse, il se proclame gestionnaire de la communauté.

L'utilisation de messages de rafraîchissement permet aux agents de détecter la disparition du gestionnaire. Un nouveau gestionnaire est alors élu en utilisant une métrique, renseignant sur la qualité du pair (ressources matérielles, comportement, ...), que l'on utilise pour déterminer un temps d'attente. Elle permet au meilleur pair de s'annoncer en premier comme nouveau gestionnaire de la communauté.

#### 4. Mise en œuvre de notre proposition sur JXTA

L'ensemble du travail de mise en œuvre de notre modèle de l'information a été effectué sur l'implantation java de la plate-forme JXTA. Nous avons intégré dans chaque pair un agent de gestion construit autour d'un agent JMX (*Java Management eXtension*) qui héberge des objets gérés et les rend accessibles par le biais du protocole RMI. Un gestionnaire, exécuté lui aussi sur un pair, est élu pour chaque communauté, appelée *Peer group* dans le contexte de JXTA. Son rôle est d'agréger les différentes données de gestion issues des agents et d'en fournir une vue synthétique qui suit le modèle de l'information que nous avons conçu. Les objets gérés qu'il génère sont aussi enregistrés dans un agent JMX, accessible par RMI. Enfin, une application de gestion qui interagit avec le gestionnaire offre à un utilisateur une véritable interface de supervision pour la plate-forme. Afin d'assurer l'indépendance entre le plan de supervision et le plan de service, nous utilisons la technologie RMI plutôt que les *pipes* JXTA.

#### 4.1. *Instanciation du modèle de l'information*

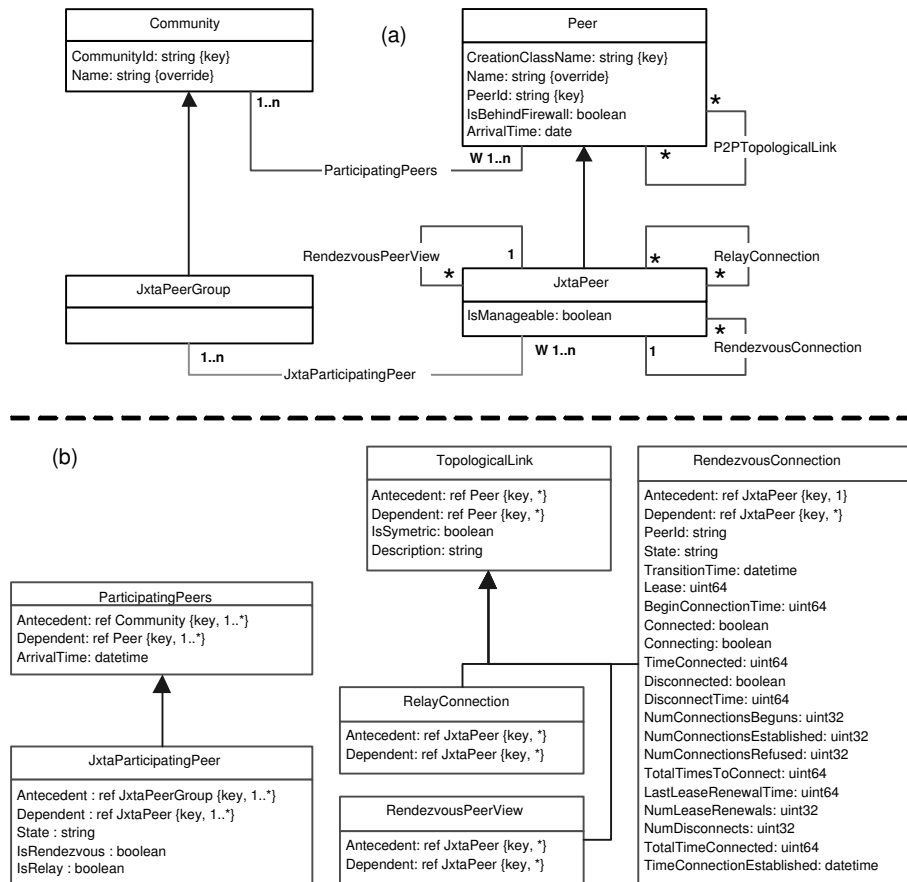
Nous avons spécialisé notre modèle de l'information générique afin d'intégrer les concepts relatifs à la plate-forme. L'ensemble des métriques inscrites sont issues de l'instrumentation faite par le projet MMP (*Metering and Monitoring Project*), un projet de la communauté JXTA qui propose de concevoir et mettre en œuvre une infrastructure de supervision pour l'implantation Java de la plate-forme ; ce travail étant toutefois incomplet et reposant exclusivement sur l'utilisation des concepts de JXTA pour sa propre supervision.

L'extension du sous-modèle de l'organisation que nous avons conçue, que nous appelons par la suite "vue globale" est représentée sur la figure 2.a. Tout d'abord, elle montre que nous avons étendu les notions génériques de pair et de communauté à travers les classes *Jxta\_JxtaPeer* et *Jxta\_JxtaPeergroup* qui correspondent aux concepts introduits par JXTA. Le rôle d'un pair dans une communauté est décrit à l'aide de la classe d'association *Jxta\_JxtaParticipatingPeer*, qui hérite de la classe *P2P\_ParticipatingPeers* et dont les attributs sont mentionnés sur la figure 2.b. Ensuite, pour représenter les relations topologiques entre les pairs, nous avons conçu trois classes d'associations qui héritent toutes de la classe générique *P2P\_TopologicalLink* : la classe *Jxta\_RendezvousConnection* représente un lien entre un pair simple et un pair de rendez-vous, la classe *Jxta\_RelayConnection*, un lien entre un pair et un relais, et enfin la classe *Jxta\_RendezvousPeerView*, un lien entre deux pairs de rendez-vous. Par le biais de ce modèle, nous sommes capables de représenter les différents pairs JXTA, les groupes et leur organisation hiérarchique, ainsi que les différentes relations topologiques que les pairs entretiennent.

#### 4.2. *Intégration d'un agent de gestion dans les pairs*

##### 4.2.1. *Définition de la vue locale*

La vue locale du modèle de l'information pour JXTA est très similaire à la vue globale. Nous ne soulignons donc ici que la manière dont nous avons traduit des classes partagées entre des pairs en pointeurs. Pour cela, nous utilisons l'exemple de la figure 3 qui illustre le cas de la représentation de la connexion d'un pair simple à son pair de rendez-vous. La figure 3.a montre qu'au niveau du modèle global, la classe *Jxta\_RendezvousConnection* lie un pair de rendez-vous à lui-même. Au niveau de l'instanciation du modèle en Java, on trouvera donc un objet de la classe *Jxta\_JxtaPeer* représentant un pair simple, un objet de la même classe représentant son pair de rendez-vous, et un objet de la classe *Jxta\_RendezvousConnection* contenant un pointeur vers chacun des deux objets cités précédemment. Les agents JMX du pair simple et du pair de rendez-vous stockeront les instances respectives de la classe *Jxta\_JxtaPeer*. La question est alors : *Où est stockée l'instance de la classe Jxta\_RendezvousConnection ?* Pour résoudre ce problème, nous proposons dans notre vue locale de définir la notion d'objet distant. Dans le cas d'un pair lié par le biais d'une association topologique, nous avons défini la classe *JxtaLocal\_RemotePeer* qui



**Figure 2.** (a) Le sous-modèle de l'organisation de JXTA – (b) Les classes d'association du sous-modèle de l'organisation

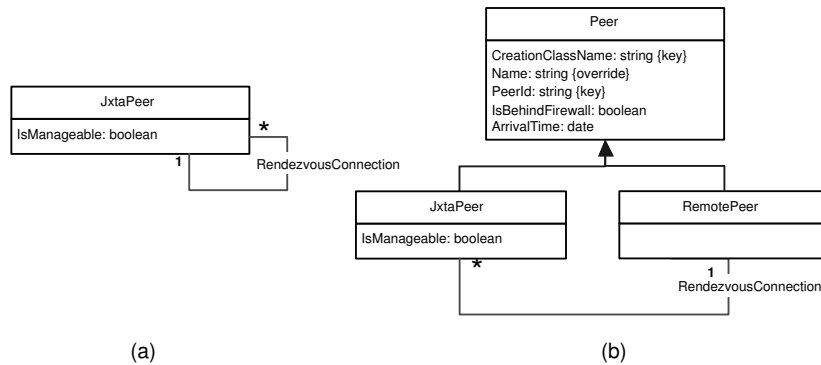
représente un pair distant. Le schéma local de cette association est représenté sur la figure 3.b. L'ensemble des objets est ainsi hébergeable au sein de l'agent JMX implanté dans le pair concerné.

#### 4.2.2. Traduction des classes MOF en classes Java

Les modèles de l'information local et global que nous avons conçus sont spécifiés sous forme de diagrammes UML que nous avons traduits dans le langage MOF (*Managed Object Format*) de CIM. Afin de traduire cette spécification MOF en classes Java, nous avons utilisé l'outil MOF2MBean<sup>2</sup> de Sun. Cet outil permet de générer

2. <http://java.sun.com/products/JavaManagement/JMXperience.html>





**Figure 3.** Exemple d'adaptation de notre modèle de l'information en une vue locale. (a) Vue globale. (b) Vue locale.

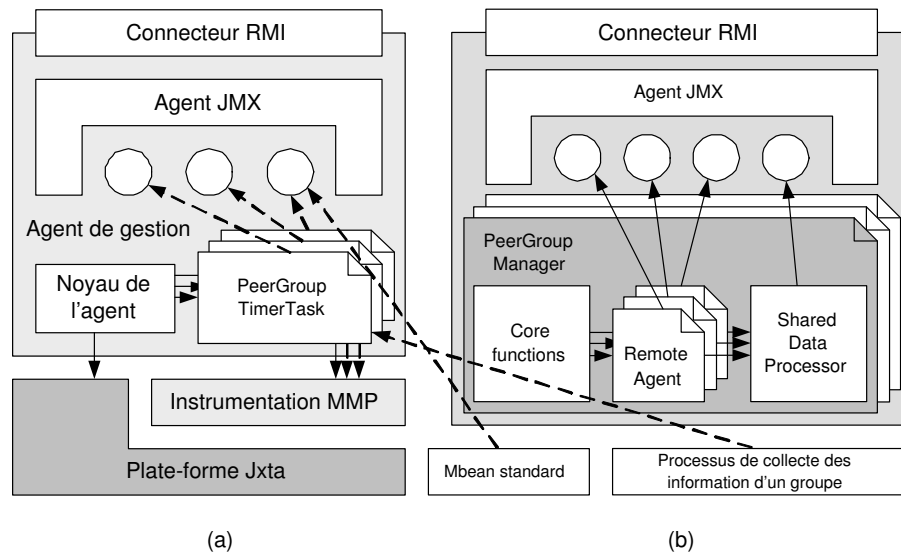
des paquetages de classes qui sont organisées de manière à respecter les concepts de CIM : chaque classe CIM génère une classe Java, l'héritage est respecté et certains qualifieurs sont adaptés à Java.

MOF2MBean implante dans chaque classe Java une interface de type MBean standard qui permet de les intégrer dans un agent JMX. Un schéma de nommage qui référence de manière unique chaque instance de classe dans l'agent est aussi proposé. Il se compose d'un nom de domaine qui peut être choisi arbitrairement et d'une série de couples `attribut=valeur` qui est constituée des attributs clés de la classe CIM. Un couple supplémentaire `class=...` qui renseigne sur la classe Java correspondant à l'objet est ajouté.

#### 4.2.3. Architecture et fonctionnement

L'architecture fonctionnelle des agents de gestion que nous avons insérés dans les pairs JXTA est représentée sur la figure 4.a. Le cœur de l'agent a pour rôle principal de détecter les groupes auxquels le pair participe et pour chacun de ces groupes, de lancer un processus autonome, appelé *PeerGroupTimerTask*, qui se charge de collecter et mettre à jour les informations de gestion relatives au groupe dont il s'occupe.

Chaque processus *PeerGroupTimerTask* rapatrié ses informations de gestion par une méthode de *polling*. Ensuite, afin de limiter les opérations d'enregistrement et de suppression d'objets de l'agent JMX, à chaque nouvelle scrutation, le processus *PeerGroupTimerTask* compare la vue qu'il possède avec celle qu'il vient d'obtenir. Seuls les objets qui diffèrent entre ces deux vues conduisent à des opérations d'insertion et de suppression dans l'agent.



**Figure 4.** (a) Architecture fonctionnelle des agents – (b) Architecture fonctionnelle du gestionnaire

### 4.3. Développement d'un gestionnaire

La vue offerte par notre gestionnaire suit le modèle de l'information présenté dans la section 4.1. Afin de respecter le modèle d'organisation de JXTA fondé sur des groupes, nous avons choisi de désigner, pour chaque groupe JXTA, un pair qui se voit confier le rôle de gestionnaire.

#### 4.3.1. Architecture et fonctionnement

L'architecture fonctionnelle du gestionnaire que nous avons développé est représentée sur la figure 4.b. Pour chaque groupe où le pair qui héberge ce service est élu gestionnaire, un processus appelé *PeerGroup Manager* est créé. Ce processus a pour fonction de collecter les données de gestion fournies par les différents agents du groupe et d'en fournir une vue agrégée. Pour cela, le *PeerGroup Manager* utilise trois composants. Le premier est le noyau du gestionnaire, appelé *Core*. Il répond aux requêtes émises par les pairs qui recherchent un gestionnaire, et gère l'enregistrement de nouveaux agents et leur suppression. Un agent peut en effet être supprimé du gestionnaire pour deux raisons : la première est sa notification au gestionnaire qu'il quitte la communauté ; la seconde est l'impossibilité pour le gestionnaire de se connecter à l'interface RMI qu'offre son agent JMX. Le deuxième composant du gestionnaire est le *Remote Agent*, un composant chargé de rapatrier les données de gestion fournies par un agent particulier. Le *PeerGroup Manager* instancie autant de *Remote Agents* qu'il

supervise d'agents. Le dernier composant, intitulé *Shared Data Processor* a pour rôle de gérer les données partagées entre plusieurs agents qui n'ont qu'une vue locale.

#### 4.3.2. *Rapatriement des données de gestion*

Pour construire la vue globale d'une communauté, le gestionnaire repose sur une approche hybride composée de *polling* et de délégation. Le *polling* est utilisé pour connaître la nature des objets de gestion qu'un agent héberge. Le gestionnaire peut ainsi créer ses propres objets, à l'image de ceux qu'il a découverts dans les agents. La délégation est utilisée pour récupérer la valeur des attributs de chacun des objets de gestion. En effet, les attributs des objets hébergés par le gestionnaire ne contiennent aucune valeur. Ils sont simplement des pointeurs vers les attributs des objets correspondants dans l'agent auquel ils font référence. Ainsi, lorsque qu'une application interroge l'agent JMX contenu dans le gestionnaire pour récupérer la valeur d'un attribut, celui-ci va déléguer cette requête à l'agent correspondant. Les valeurs des attributs sont ainsi les plus proches possibles de la réalité.

#### 4.4. *Développement d'une application de gestion*

Le travail d'agrégation des données de gestion effectué par le gestionnaire nous a permis de développer une application de gestion qui offre à un administrateur une interface de supervision de la plate-forme. Pour pouvoir accéder aux données hébergées par le gestionnaire d'un groupe, l'application doit découvrir son interface de communication. Pour ce faire, elle exécute un pair JXTA qui va scruter les différents groupes accessibles et chercher au sein de chaque groupe l'existence d'un gestionnaire. Chaque gestionnaire présent répond à cette requête de découverte en indiquant son adresse RMI. Une fois l'opération de recherche effectuée, le pair est arrêté.

Les fonctions offertes par notre application s'articulent principalement autour d'une vue topologique qui distingue les différents types de pairs (pairs simples, pairs de rendez-vous et relais) et les différents types de liens (connexion d'un pair à son pair de rendez-vous, entre pairs de rendez-vous et d'un pair à un pair relais). Les fonctions de surveillance permettent de s'abonner à une des métriques numérique proposées dans les données de gestion et de surveiller sa valeur, avec un tracé de son évolution ; le franchissement d'un seuil déclenchant une alarme (inscription dans un fichier journal, apparition d'un message dans l'application ou envoi d'un mail à l'administrateur). D'un point de vue de la réalisation, nous avons délégué cette tâche aux agents à travers l'utilisation des objets `monitor` de JMX. Ces objets sont capables de surveiller une métrique et d'envoyer des notifications en cas d'un franchissement de seuil. Cette délégation évite ainsi tout goulot d'étranglement au niveau de l'application.

Concernant le contrôle de la plate-forme, la seule action que nous avons rendue possible actuellement concerne, pour un pair donné, le démarrage et l'arrêt de services à distance. Cette fonctionnalité, combinée à celle de surveillance d'une métrique, permet de créer des règles de fonctionnement que la plate-forme va devoir appliquer.

## 5. Conclusion et travaux futurs

Le modèle P2P se présente actuellement comme une véritable alternative au modèle client-serveur. Néanmoins, son déploiement dans des environnements contraints comme les entreprises ou les administrations où la qualité du service offert doit être garantie passe par l'intégration de mécanismes de supervision. Les approches de gestion actuelles s'intègrent mal aux caractéristiques du modèle P2P et doivent donc être repensées.

C'est pourquoi, concernant la modélisation des informations de gestion, nous proposons d'étendre CIM pour la supervision des réseaux et services P2P. Notre proposition est générique et peut donc être instanciée sur n'importe quelle infrastructure P2P existante. Nous avons effectué ce travail sur Pastry, Chord et Jxta. Enfin notre modèle est complet en ce sens qu'il prend en compte les différentes caractéristiques du modèle : l'organisation des communautés, la topologie, la communication, les ressources, les services et le routage.

Nous avons déployé ce modèle de l'information sur Jxta. Pour ce faire, nous avons spécialisé notre modèle en y intégrant les points caractéristiques de la plate-forme et en nourrissant nos classes de métriques issues d'une instrumentation existante de Jxta. A travers une architecture de supervision centralisée, nous avons montré la manière dont un gestionnaire et une application de supervision pouvaient tirer parti de ces informations et proposer ainsi des fonctions de surveillance et de contrôle.

Notre travail actuel porte sur la valorisation de notre réalisation. La communauté Jxta est très favorable à ce travail et nous sommes en train de créer un projet Jxta qui permette à certains d'utiliser notre prototype mais aussi à d'autres de contribuer à son développement. Enfin, concernant les perspectives, nous prévoyons d'évaluer notre réalisation à travers différents scénarii de tests qui puissent, entre autres, évaluer le passage à l'échelle de notre proposition.

## 6. Bibliographie

- [ABE 01] ABERER K., DESPOTOVIC Z., « Managing trust in a peer-2-peer information system », *Proceedings of the tenth international conference on Information and Knowledge Management - CIKM'01*, ACM Press, 2001, p. 310–317.
- [ANT 02] ANTONIADIS P., COURCOUBETIS C., « Market Models for P2P Content Distribution », MORO G., KOUBARAKIS M., Eds., *Proceedings of the First International Workshop on Agents and Peer-To-Peer Computing - AP2PC'02*, n° 2530 LNCS, Springer-Verlag, 2002, p. 138–143.
- [BUM 00] BUMPUS W., SWEITZER J. W., THOMPSON P., R. W. A., WILLIAMS R. C., *Common Information Model*, Wiley, 2000.
- [BUR 03] BURAGOHAIN C., AGRAWAL D., SURI S., « A Game Theoretic Framework for Incentives in P2P Systems », *Proceedings of the 3rd International Conference on Peer-to-Peer Computing - P2P'03*, IEEE Computer Society, 2003, p. 48–57.

- [COO 05] COOPER B. F., « Ad Hoc, Self-Supervising Peer-to-Peer Search Networks », *ACM Transactions on Information Systems*, vol. 23, n° 2, 2005, p. 169–200.
- [DOY 04] DOYEN G., FESTOR O., NATAF E., « A CIM extension for peer-to-peer network and service management », DE SOUZA J., DINI P., Eds., *Proceedings of the 11th International Conference on Telecommunication - ICT'04*, n° 3124 LNCS, Springer-Verlag, 2004, p. 801–810.
- [DOY 05] DOYEN G., NATAF E., FESTOR O., « A hierarchical architecture for a distributed management of P2P networks and services », SCHÖNWÄLDER J., SERRAT J., Eds., *Proceedings of the 16th IFIP/IEEE International Workshop on Distributed Systems : Operations and Management - DSOM 2005*, n° 3775 LNCS, Springer-Verlag, 2005, p. 257–268.
- [FEL 04] FELDMAN M., LAI K., STOICA I., CHUANG J., « Robust incentive techniques for peer-to-peer networks », *Proceedings of the 5th ACM conference on Electronic Commerce - EC'04*, ACM Press, 2004, p. 102–111.
- [GOL 01] GOLLE P., LEYTON-BROWN K., MIRONOV I., LILLIBRIDGE M., « Incentives for sharing in peer-to-peer networks », *Proceedings of the 3rd ACM conference on Electronic Commerce - EC'01*, ACM Press, 2001, p. 264–267.
- [GON 01] GONG L., « JXTA : A network programming environment », *IEEE Internet Computing*, vol. 5, n° 3, 2001, p. 88–95.
- [GUP 03] GUPTA M., JUDGE P., AMMAR M., « A reputation system for peer-to-peer networks », *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video - NOSSDAV'03*, ACM Press, 2003, p. 144–152.
- [KON 03] KONSTANTINOOU A. V., « Towards autonomic networks », PhD thesis, Columbia university, 2003.
- [KOU 03] KOULOURIS T., HENJES R., TUTSCHKU K., DE MEER H., « Implementation of Adaptive Control for P2P Overlays. », WAKAMIYA N., SOLARSKI M., STERBENZ J. P. G., Eds., *Proceedings of the 5th IFIP TC6 International Workshop on Active Networks - IWAN'03*, n° 2982 LNCS, Springer-Verlag, 2003, p. 292–306.
- [ROW 01] ROWSTRON A., DRUSCHEL P., « Pastry : Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems », *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms - Middleware'01*, n° 2218 LNCS, Springer-Verlag, 2001, p. 329–350.
- [SCH 03] SCHECHTER S. E., GREENSTADT R. A., SMITH M. D., « Trusted Computing, Peer-To-Peer Distribution, and the Economics of Pirated Entertainment », *Proceedings of the Second Workshop on Economics and Information Security*, 2003.
- [STO 01] STOICA I., MORRIS R., KARGER D., KAASHOEK M. F., BALAKRISHNAN H., « Chord : A scalable peer-to-peer lookup service for internet applications », *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication - SIGCOMM'01*, ACM Press, 2001, p. 149–160.
- [VOU 05] VOULGARIS S., GAVIDIA D., VAN STEEN M., « CYCLON : Inexpensive Membership Management for Unstructured P2P Overlays », *Journal of Network and Systems Management*, vol. 13, n° 2, 2005, p. 197–217.
- [XIO 02] XIONG L., LIU L., « Building Trust in Decentralized Peer-to-Peer Electronic Communities », *International Conference on Electronic Commerce Research - ICECR-5*, 2002.